D 7

```
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  LLL              EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  EEEEEEEEEEEEEEE
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  LLL              EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  EEEEEEEEEEEEEEE
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  LLL              EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  EEEEEEEEEEEEEEE
DDD         DDD  EEE              LLL              EEE                    TTT          EEE
DDD         DDD  EEE              LLL              EEE                    TTT          EEE
DDD         DDD  EEE              LLL              EEE                    TTT          EEE
DDD         DDD  EEE              LLL              EEE                    TTT          EEE
DDD         DDD  EEE              LLL              EEE                    TTT          EEE
DDD         DDD  EEE              LLL              EEE                    TTT          EEE
DDD         DDD  EEEEEEEEEEEE     LLL              EEEEEEEEEEEE           TTT          EEEEEEEEEEEE
DDD         DDD  EEEEEEEEEEEE     LLL              EEEEEEEEEEEE           TTT          EEEEEEEEEEEE
DDD         DDD  EEEEEEEEEEEE     LLL              EEEEEEEEEEEE           TTT          EEEEEEEEEEEE
DDD         DDD  EEE              LLL              EEE                    TTT          EEE
DDD         DDD  EEE              LLL              EEE                    TTT          EEE
DDD         DDD  EEE              LLL              EEE                    TTT          EEE
DDD         DDD  EEE              LLL              EEE                    TTT          EEE
DDD         DDD  EEE              LLL              EEE                    TTT          EEE
DDD         DDD  EEE              LLL              EEE                    TTT          EEE
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  LLLLLLLLLLLLLLL  EEEEEEEEEEEEEEE        TTT          EEEEEEEEEEEEEEE
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  LLLLLLLLLLLLLLL  EEEEEEEEEEEEEEE        TTT          EEEEEEEEEEEEEEE
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  LLLLLLLLLLLLLLL  EEEEEEEEEEEEEEE        TTT          EEEEEEEEEEEEEEE
```

```
**FILE**ID**DELEMAIN                                D 8




DDDDDDDD    EEEEEEEEEE  LL          EEEEEEEEEE  MM      MM    AAAAAA      IIIIII    NN      NN
DDDDDDDD    EEEEEEEEEE  LL          EEEEEEEEEE  MM      MM    AAAAAA      IIIIII    NN      NN
DD      DD  EE          LL          EE          MMMM  MMMM  AA      AA      II      NN      NN
DD      DD  EE          LL          EE          MMMM  MMMM  AA      AA      II      NN      NN
DD      DD  EE          LL          EE          MM  MM  MM  AA      AA      II      NNNN    NN
DD      DD  EE          LL          EE          MM  MM  MM  AA      AA      II      NNNN    NN
DD      DD  EEEEEEEE    LL          EEEEEEEE    MM      MM  AA      AA      II      NN  NN  NN
DD      DD  EEEEEEEE    LL          EEEEEEEE    MM      MM  AA      AA      II      NN  NN  NN
DD      DD  EE          LL          EE          MM      MM  AAAAAAAAAA      II      NN    NNNN
DD      DD  EE          LL          EE          MM      MM  AAAAAAAAAA      II      NN    NNNN
DD      DD  EE          LL          EE          MM      MM  AA      AA      II      NN      NN      ....
DD      DD  EE          LL          EE          MM      MM  AA      AA      II      NN      NN      ....
DDDDDDDD    EEEEEEEEEE  LLLLLLLLLL  EEEEEEEEEE  MM      MM  AA      AA    IIIIII    NN      NN      ....
DDDDDDDD    EEEEEEEEEE  LLLLLLLLLL  EEEEEEEEEE  MM      MM  AA      AA    IIIIII    NN      NN      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE delemain (          ! STARLET Native File Deletion Utility
0                            IDENT = 'V04-000',
0                            MAIN = del$main,
0                            ADDRESSING_MODE(EXTERNAL=GENERAL)
0                            ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 ! FACILITY:  DELETE
1 !
1 ! ABSTRACT:
1 !
1 !       This utility program deletes one or more user-specified files.
1 !
1 ! ENVIRONMENT:
1 !
1 !       VAX/VMS operating system, unprivileged user mode utility,
1 !       operates at non-AST level.
1 !
1 ! AUTHOR:  Stephen H. Zalewski,          CREATION DATE:  10-Aug-1982
1 !
1 ! Modified by:
1 !
1 !       V03-011 SHZ0014         Stephen H. Zalewski,    15-Mar-1984
1 !               Modify $PARSE call to parse with no I/O.  Move all wild card
1 !               defaulting for PURGE into this module.
1 !
1 !       V03-010 SHZ0013         Stephen H. Zalewski,    20-Feb-1984
1 !               Add support for sticky searchlists.
1 !
1 !       V03-009 SHZ0012         Stephen H. Zalewski,    27-Dec-1983
1 !               Move the defaulting of file name and type for $PURGE into
1 !               module PURGE.  Make all RMS structures LOCAL instead of GLOBAL.
```

```
 1 !
 1 !      V03-008 SHZ0011         Stephen H. Zalewski,    31-Oct-1983
 1 !              Do not modify pointer to new related name related name block.
 1 !              This is now done in lib$file_scan.
 1 !
 1 !      V03-007 SHZ0010         Stephen H. Zalewski,    25-Feb-1983
 1 !              If user issued a PURGE command with the /LOG qualifier,
 1 !              and no files were purged, issue a message to the user
 1 !              reporting this.
 1 !
 1 !      V03-006 SHZ0009         Stephen H. Zalewski,    11-Feb-1983
 1 !              Changed lib$v_cqf_uic to lib$v_cqf_byowner.
 1 !
 1 !      V03-005 SHZ0008         Stephen H. Zalewski,    29-Oct-1982  20:46
 1 !              Placed all routines formerly residing in DELSPECS.B32 into
 1 !              this module.  Modified delete to use the common qualifier
 1 !              library package.  Use lib$file_scan to do all wildcard
 1 !              processing.
 1 !
 1 !      V03-004 SHZ0007         Stephen H. Zalewski,    6-Sep-1982  18:53
 1 !              Fix bug that incorrectly reported block size of a file.
 1 !
 1 !      V03-003 SHZ0006         Stephen H. Zalewski,    26-Aug-1982  22:23
 1 !              Backout defaulting of version number for DELETE.  Fix bug
 1 !              introduced in SHZ0005 than prevented dangling directory
 1 !              entries from being deleted if the DELETE command was
 1 !              issued with /LOG qualifier.  Finally, if a file is opened
 1 !              because of /SINCE, /BEFORE or /LOG qualifiers being present,
 1 !              leave it open until we actually delete it to optimize
 1 !              the number of FAL jobs necessary to do the job in case
 1 !              we are doing this over the net.
 1 !
 1 !      V03-002 SHZ0005         Stephen H. Zalewski,    10-Aug-1982  21:04
 1 !              Extensive rewrite to use new CLI.  Wild card specs are now
 1 !              supported for /ERASE qualifer.  Info messages for /LOG
 1 !              now contain block size of file and total files deleted/purged
 1 !              are now printed.  Rewrote the way /CONFIRM is done.  Rewrote
 1 !              condition handler.  Explicit version number is no longer
 1 !              needed for DELETE (default is now ";0").  Rewrote error
 1 !              handling routine(s).
 1 !
 1 !      V03-001 SHZ0004         Stephen H. Zalewski,    23-Mar-1982  14:30
 1 !              Have correct version number displayed when /CONFIRM is
 1 !              used and version number is negative or zero.
 1 !
 1 !--
```

```
;  105        0104   1 LIBRARY 'SYS$LIBRARY:STARLET.L32';                       ! VAX/VMS common definitions
;  106        0105   1 REQUIRE 'SRC$:DELETE.REQ';                               ! COPY literal definitions and macros
;  107        0211   1
;  108        0212   1
;  109        0213   1 FORWARD ROUTINE
;  110        0214   1     del$main,                                            ! Main DELETE control routine
;  111        0215   1     delete_file          : NOVALUE,                      ! Deletes one file
;  112        0216   1     del$get_cmdqual      : NOVALUE,                      ! Get command line and parse it
;  113        0217   1     del$get_file,                                        ! Obtains the input file specification and parses it
;  114        0218   1     del$search_error:   NOVALUE,                         ! Error searching for a file.
;  115        0219   1     condit_handler,                                      ! Condition handler for errors and messages
;  116        0220   1     del$file_error       : NOVALUE;                      ! Handles errors on RMS file functions
;  117        0221   1
;  118        0222   1
;  119        0223   1 EXTERNAL ROUTINE
;  120        0224   1     purge_files,                                         ! Purges files related to one file specification
;  121        0225   1     purge_ods1_directory,                                ! Purge an ODS-1 directory lost.
;  122        0226   1     cli$get_value,                                       ! Get qualifier value
;  123        0227   1     cli$present,                                         ! Determine if qualifier is present
;  124        0228   1     lib$cvt_dtb,                                         ! Converts an ASCII string to an integer value.
;  125        0229   1     lib$file_scan,                                       ! Search wildcard specifications.
;  126        0230   1     lib$qual_file_match,                                 ! Check to see if a file should be deleted
;  127        0231   1     lib$qual_file_parse,                                 ! Get common command qualifiers
;  128        0232   1     lib$set_erase;                                       ! Mark a file for erase-on-delete
;  129        0233   1
;  130        0234   1
;  131        0235   1 EXTERNAL
;  132        0236   1     version_list,
;  133        0237   1     lib$_filfaimat,                                      ! File failed to meet criteria
;  134        0238   1     lib$_quipro;                                         ! Quit processing
;  135        0239   1
;  136        0240   1
;  137        0241   1 GLOBAL
;  138        0242   1     scan_context         : INITIAL(0),
;  139        0243   1     del$context,                                         ! Context parameter for RTL routines.
;  140        0244   1     del$cli_status       : $BBLOCK [4] INITIAL (0),      ! DELETE status word used by CLI interface
;  141        0245   1     del$keepver_val      : INITIAL (1),                  ! Number of versions to keep during a PURGE.
;  142        0246   1     del$files_deleted    : INITIAL (0),                  ! Total number of files purged or deleted
;  143        0247   1     del$blocks_deleted,                                  ! Total number of blocks deleted
;  144        0248   1     del$file_size,                                       ! Block size of file being deleted
;  145        0249   1     del$exit_status      : $BBLOCK [4]                   ! Holds most severe error code
;  146        0250   1                            INITIAL (ss$_normal),
;  147        0251   1     infile_desc          : $BBLOCK [dsc$c_s_bln];        ! CLI input file descriptor block
;  148        0252   1
;  149        0253   1
;  151          0254  1 ROUTINE del$main =
;  152          0255  1
;  153          0256  1 !++
;  154          0257  1 ! Functional description
;  155          0258  1 !
;  156          0259  1 !       This routine is the central control routine for the DELETE utility.
;  157          0260  1 !       It determines the basic logical flow and calls support routines
;  158          0261  1 !       that perform each logical function in deleting and purging files.
;  159          0262  1 !
;  160          0263  1 ! Calling sequence
;  161          0264  1 !
;  162          0265  1 !       del$main ()
;  163          0266  1 !
;  164          0267  1 ! Input parameters
;  165          0268  1 !
;  166          0269  1 !       none
;  167          0270  1 !
;  168          0271  1 ! Implicit inputs
;  169          0272  1 !
;  170          0273  1 !       none
;  171          0274  1 !
;  172          0275  1 ! Output parameters
;  173          0276  1 !
;  174          0277  1 !       none
;  175          0278  1 !
;  176          0279  1 ! Implicit outputs
;  177          0280  1 !
;  178          0281  1 !       del$exit_status - set whenever an error occurs
;  179          0282  1 !
;  180          0283  1 ! Routine value
;  181          0284  1 !
;  182          0285  1 !       Most severe error encountered during processing or SS$_NORMAL
;  183          0286  1 !
;  184          0287  1 ! Side effects
;  185          0288  1 !
;  186          0289  1 !       The specified files are deleted.
;  187          0290  1 !
;  188          0291  1 !--
;  189          0292  1
;  190          0293  2 BEGIN
;  191          0294  2
;  192          0295  2 LOCAL
;  193          0296  2     infile_name         : VECTOR [nam$c_maxrss, BYTE],  ! File name after open
;  194          0297  2     infile_xname        : VECTOR [nam$c_maxrss, BYTE],  ! File name before open
;  195          0298  2     infile_fab          : $FAB_DECL,                    ! Space for the file FAB
;  196        P 0299  2     infile_nam_blk      : $NAM (                        ! File name block:
;  197        P 0300  2                             RSA = infile_name,          !    Address and length of the
;  198        P 0301  2                             RSS = nam$c_maxrss,         !       file name after open
;  199        P 0302  2                             ESA = infile_xname,         !    Address and length of the
;  200          0303  2                             ESS = nam$c_maxrss),        !       file name before open
;  201          0304  2     infile_xabdat       : $XABDAT (),                   !    Date/time XAB
;  202          0305  2     infile_xabpro       : $XABPRO (NXT = infile_xabdat),!    Protection XAB (date/time XAB chained to this X
;  203          0306  2     status;                                         ! General routine return code
;  204          0307  2
;  205          0308  2 LABEL
;  206          0309  2     process_loop;                                   ! Label for file processing loop
;  207          0310  2
```

```
2
2 ENABLE condit_handler;                                   ! Enable a local handler.
2
2     del$get_cmdqual ();                                  ! Get the command qualifiers.
2
2  !
2  ! The remainder of this routine is executed for each
2  ! file specification supplied by the user.
2  !
2
2     WHILE true DO                                        ! Beginning of repeat loop
2 process_loop:
2         BEGIN
3         status = del$get_file ( infile_fab,              ! Get a file spec.  Pass the fab,
3                                 infile_nam_blk,          !    the address of the NAM block,
3                                 infile_xabpro);          !    and the address of the XAB chain.
3         IF .status EQL no_more_files                     ! If there are no more file specs,
3             THEN EXITLOOP;                               !    then exit the file spec processing loop.
3         IF .status EQL bad_version                       ! If bad version specified
3             THEN EXITLOOP;                               !    then quit.
3         IF NOT .status                                   ! If the file specification wasn't okay,
3             THEN LEAVE process_loop;                     !    then go get the next one.
3
3  !
3  ! Perform DELETE or PURGE processing for this file specification.
3  !
3
3         IF .del$cli_status [del$v_purge_cmd]             ! If this is a PURGE command
3             THEN  lib$file_scan(infile_fab,              !  then call purge routine with addresses of the FAB
3                                 purge_files,
3                                 del$search_error,
3                                 scan_context)
3             ELSE  lib$file_scan(infile_fab,              !  otherwise delete the files.
3                                 delete_file,
3                                 del$search_error,
3                                 scan_context);
3
3         IF .del$cli_status [del$v_cntrl_z_stop]          ! If the user said <control-Z> to a delete confirmat
3             THEN EXITLOOP;                               ! then, stop
3
3         IF .version_list NEQ 0                           ! Make sure all ODS-1 files are purged.
3             THEN purge_ods1_directory(version_list);     ! Delete old versions
3
2         END;                                             ! End of "WHILE 1 DO" file spec processing loop.
2
2  !
2  !  If this was a PURGE, no files were purge, and the user said /LOG, then
2  !  notify the user.
2  !
2
2     IF .del$cli_status[del$v_purge_cmd] AND              ! If this was a PURGE command
2       (.del$files_deleted EQL 0) AND                     !  and no files purged for this file spec,
2        .del$cli_status[del$v_log_msg]                    !  and /LOG was set
2     THEN                                                 !  then
2         put_message(msg$_nofilpurg,1,$descriptor('?')); !  output message saying so
2
2
```

```
;   265           0368  2 !
;   266           0369  2 !  If more than one file was DELETED or PURGED, and the user said /LOG, then
;   267           0370  2 !  notify the user of the number of files deleted/purged.
;   268           0371  2 !
;   269           0372  2     IF .del$cli_status [del$v_log_msg] AND               ! If /LOG qualifier present
;   270           0373  3        (.del$files_deleted GTR 1)                       !  and more thant one file deleted,
;   271           0374  2     THEN
;   272         P 0375  2        put_message (msg$_total,2,.del$files_deleted,    !  then print total number of files deleted.
;   273           0376  2                .del$blocks_deleted);
;   274           0377  2
;   275           0378  2
;   276           0379  2 RETURN .del$exit_status OR sts$m_inhib_msg;             ! EXIT with no message
;   277           0380  1 END;


                                                   .TITLE  DELEMAIN
                                                   .IDENT  \V04-000\

                                                   .PSECT  $PLIT$,NOWRT,NOEXE,2

                         02  00000 P.AAA:  .BYTE   2
                         60  00001         .BYTE   96
                         FF  00002         .BYTE   -1
                         00  00003         .BYTE   0
                   00000000  00004         .LONG   0
                         00  00008         .BYTE   0
                         00  00009         .BYTE   0
                         FF  0000A         .BYTE   -1
                         00  0000B         .BYTE   0
                   00000000  0000C         .LONG   0
                   00000000  00010         .LONG   0
                      0000#  00014         .WORD   0[8]
                      0000#  00024         .WORD   0[3]
                      0000#  0002A         .WORD   0[3]
                   00000000  00030         .LONG   0
                   00000000  00034         .LONG   0
                         00  00038         .BYTE   0
                         00  00039         .BYTE   0
                         00  0003A         .BYTE   0
                         00  0003B         .BYTE   0
                         00  0003C         .BYTE   0
                         00  0003D         .BYTE   0
                        00#  0003E         .BYTE   0[2]
                   00000000  00040         .LONG   0
                   00000000  00044         .LONG   0
                   00000000  00048         .LONG   0
                   00000000  0004C         .LONG   0
                   00000000  00050         .LONG   0
                   00000000  00054         .LONG   0
                  00000000#  00058         .LONG   0[2]
                         12  00060 P.AAB:  .BYTE   18
                         2C  00061         .BYTE   44
                       0000  00062         .WORD   0
                   00000000  00064         .LONG   0
                       0000  00068         .WORD   0
                       0000  0006A         .WORD   0
                  00000000#  0006C         .LONG   0[2]
```

```
 00000000# 00074          .LONG    0[2]                          ;
 00000000  0007C          .LONG    0                             ;
 00000000  00080          .LONG    0                             ;
 00000000# 00084          .LONG    0[2]                          ;
        13 0008C P.AAC:   .BYTE    19                            ;
        58 0008D          .BYTE    88                            ;
      0000 0008E          .WORD    0                             ;
  00000000 00090          .LONG    0                             ;
      FFFF 00094          .WORD    -1                            ;
        00 00096          .BYTE    0                             ;
        00 00097          .BYTE    0                             ;
0000  0000 00098          .WORD    0, 0                          ;
        00 0009C          .BYTE    0                             ;
        00 0009D          .BYTE    0                             ;
      0000 0009E          .WORD    0                             ;
  00000000 000A0          .LONG    0                             ;
  00000000 000A4          .LONG    0                             ;
      0000 000A8          .WORD    0                             ;
      0000 000AA          .WORD    0                             ;
  00000000 000AC          .LONG    0                             ;
  00000000 000B0          .LONG    0                             ;
        3F 000B4 P.AAE:   .ASCII   \?\                           ;
           000B5          .BLKB    3
 00000001  000B8 P.AAD:   .LONG    1                             ;
 00000000' 000BC          .ADDRESS P.AAE                         ;

                          .PSECT  $GLOBAL$,NOEXE,2

 00000000  00000 SCAN_CONTEXT::
                          .LONG    0                             ;
           00004 DEL$CONTEXT::
                          .BLKB    4
 00000000  00008 DEL$CLI_STATUS::
                          .LONG    0                             ;
 00000001  0000C DEL$KEEPVER_VAL::
                          .LONG    1                             ;
 00000000  00010 DEL$FILES_DELETED::
                          .LONG    0                             ;
           00014 DEL$BLOCKS_DELETED::
                          .BLKB    4
           00018 DEL$FILE_SIZE::
                          .BLKB    4
 00000001  0001C DEL$EXIT_STATUS::
                          .LONG    1                             ;
           00020 INFILE_DESC::
                          .BLKB    8

                          .EXTRN   PURGE_FILES, PURGE_ODS1_DIRECTORY
                          .EXTRN   CLI$GET_VALUE, CLI$PRESENT
                          .EXTRN   LIB$CVT_DTB, LIB$FILE_SCAN
                          .EXTRN   LIB$QUAL_FILE_MATCH
                          .EXTRN   LIB$QUAL_FILE_PARSE
                          .EXTRN   LIB$SET_ERASE, VERSION_LIST
                          .EXTRN   LIB$_FILFAIMAT, LIB$_QUIPRO

                          .PSECT  $CODE$,NOWRT,2
```

```
                                                            01FC 00000 DEL$MAIN:
                                                                                .WORD    Save R2,R3,R4,R5,R6,R7,R8                    ; 0254
                                          58  00000000G   00  9E 00002          MOVAB    LIB$SIGNAL, R8                               ;
                                          57  00000000G   00  9E 00009          MOVAB    VERSION_LIST, R7                             ;
                                          56      0000'   CF  9E 00010          MOVAB    DEL$CLI_STATUS, R6                           ;
                                          5E       FCCC   CE  9E 00015          MOVAB    -820(SP), SP                                 ;
                    0084    CE      0000'   CF       0060   8F  28 0001A          MOVC3    #96, P.AAA, INFILE_NAM_BLK                   ; 0303
                                    0088   CE       FF00   CD  9E 00024          MOVAB    INFILE_NAME, INFILE_NAM_BLK+4                ; 0293
                                    0090   CE       0134   CE  9E 0002B          MOVAB    INFILE_XNAME, INFILE_NAM_BLK+12              ;
                      58    AE      0000'   CF              2C  28 00032          MOVC3    #44, P.AAB, INFILE_XABDAT                    ; 0304
    0058   BF                00      0000'   CF              28  2C 00039          MOVC5    #40, P.AAC, #0, #88, INFILE_XABPRO           ; 0305
                                                            6E     00042                                                               ;
                                      04   AE         58   AE  9E 00043          MOVAB    INFILE_XABDAT, INFILE_XABPRO+4               ; 0304
                                           6D       00A0   CF  DE 00048          MOVAL    7$, (FP)                                     ; 0305
                                   0000V   CF              00  FB 0004D          CALLS    #0, DEL$GET_CMDQUAL                          ; 0314
                                                            5E  DD 00052 1$:      PUSHL    SP                                           ; 0324
                                                    0088   CE  9F 00054          PUSHAB   INFILE_NAM_BLK                               ;
                                                    00EC   CE  9F 00058          PUSHAB   INFILE_FAB                                   ;
                                   0000V   CF              03  FB 0005C          CALLS    #3, DEL$GET_FILE                             ;
                                           52              50  D0 00061          MOVL     R0, STATUS                                   ;
                                           03              52  D1 00064          CMPL     STATUS, #3                                   ; 0327
                                                            44  13 00067          BEQL     4$                                           ;
                                           02              52  D1 00069          CMPL     STATUS, #2                                   ; 0329
                                                            3F  13 0006C          BEQL     4$                                           ;
                                           E1              52  E9 0006E          BLBC     STATUS, 1$                                   ; 0331
                            0F             66              03  E1 00071          BBC      #3, DEL$CLI_STATUS, 2$                       ; 0338
                                                      F8   A6  9F 00075          PUSHAB   SCAN_CONTEXT                                 ; 0339
                                                   0000V   CF  9F 00078          PUSHAB   DEL$SEARCH_ERROR                             ;
                                               00000000G   00  9F 0007C          PUSHAB   PURGE_FILES                                  ;
                                                            0B  11 00082          BRB      3$                                           ;
                                                      F8   A6  9F 00084 2$:      PUSHAB   SCAN_CONTEXT                                 ; 0343
                                                   0000V   CF  9F 00087          PUSHAB   DEL$SEARCH_ERROR                             ;
                                                   0000V   CF  9F 0008B          PUSHAB   DELETE_FILE                                  ;
                                                    00F0   CE  9F 0008F 3$:      PUSHAB   INFILE_FAB                                   ;
                               00000000G   00              04  FB 00093          CALLS    #4, LIB$FILE_SCAN                            ;
                            0F             66              05  E0 0009A          BBS      #5, DEL$CLI_STATUS, 4$                       ; 0348
                                                            67  D5 0009E          TSTL     VERSION_LIST                                 ; 0351
                                                            B0  13 000A0          BEQL     1$                                           ;
                                                            57  DD 000A2          PUSHL    R7                                           ; 0352
                               00000000G   00              01  FB 000A4          CALLS    #1, PURGE_ODS1_DIRECTORY                     ;
                                                            A5  11 000AB          BRB      1$                                           ; 0321
                            18             66              03  E1 000AD 4$:      BBC      #3, DEL$CLI_STATUS, 5$                       ; 0361
                                                      08   A6  D5 000B1          TSTL     DEL$FILES_DELETED                            ; 0362
                                                            13  12 000B4          BNEQ     5$                                           ;
                            28             66              01  E1 000B6          BBC      #1, DEL$CLI_STATUS, 6$                       ; 0363
                                                   0000'   CF  9F 000BA          PUSHAB   P.AAD                                        ; 0365
                                                            01  DD 000BE          PUSHL    #1                                           ;
                                                0093122B   8F  DD 000C0          PUSHL    #9638443                                     ;
                                           68              03  FB 000C6          CALLS    #3, LIB$SIGNAL                               ;
                            15             66              01  E1 000C9 5$:      BBC      #1, DEL$CLI_STATUS, 6$                       ; 0372
                                           01         08   A6  D1 000CD          CMPL     DEL$FILES_DELETED, #1                        ; 0373
                                                            0F  15 000D1          BLEQ     6$                                           ;
                                           7E         08   A6  7D 000D3          MOVQ     DEL$FILES_DELETED, -(SP)                     ; 0376
                                                            02  DD 000D7          PUSHL    #2                                           ;
                                                00931313   8F  DD 000D9          PUSHL    #9638675                                     ;
                                           68              04  FB 000DF          CALLS    #4, LIB$SIGNAL                               ;
                            50        14   A6   10000000   8F  C9 000E2 6$:      BISL3    #268435456, DEL$EXIT_STATUS, R0              ; 0379
```

```
                                          04 000EB         RET                                   ; 0380
                                        0000 000EC  7$:    .WORD    Save nothing                 ; 0305
                                     7E   D4 000EE         CLRL     -(SP)                        ;
                                     5E   DD 000F0         PUSHL    SP                           ;
                       7E       04   AC   7D 000F2         MOVQ     4(AP), -(SP)                 ;
                 0000V CF            03   FB 000F6         CALLS    #3, CONDIT_HANDLER           ;
                                          04 000FB         RET                                   ;

; Routine Size:  252 bytes,    Routine Base:  $CODE$ + 0000


;  278          0381  1
```

```
1 ROUTINE delete_file (fab_block) : NOVALUE =
1
1 !++
1 ! Functional description
1 !
1 !       This routine deletes one file.
1 !
1 !       If the user specified the /CONFIRM qualifier, the name of the file is output to
1 !       SYS$OUTPUT and the user can advise whether the file is to be deleted or not.
1 !
1 ! Calling sequence
1 !
1 !       delete_file (fab_block.ra.v)
1 !
1 ! Input parameters
1 !
1 !       fab_block       - The FAB block for the file specification
1 !
1 ! Implicit inputs
1 !
1 !       Bits are tested in the status word:
1 !
1 !               LOG_MSG         - whether each deletion is to be logged on SYS$OUTPUT.
1 !
1 !       Fields in the NAM block are used to log the deletion:
1 !
1 !               RSA     - address of the resultant name string
1 !               RSL     - length of the resultant name string
1 !
1 ! Output parameters
1 !
1 !       none
1 !
1 ! Implicit outputs
1 !
1 !       none
1 !
1 ! Routine value
1 !
1 !       novalue
1 !
1 ! Side effects
1 !
1 !       Errors are signaled.  Files are deleted.
1 !
1 !--
2     BEGIN
2
2     MAP
2       fab_block       : REF $BBLOCK;
2
2     BIND
2       nam_block = .fab_block [fab$l_nam]  : $BBLOCK;  ! Associated NAM block address.
2
2     LOCAL
2       prompt_desc : VECTOR [2],                       ! String descriptor for prompt args.
2       name_desc : VECTOR [2],                         ! String descriptor for file name.
```

```
        status;                                        ! Holds RMS status codes.


    IF .del$cli_status [del$v_cntrl_z_stop]            ! If user has responed to /CONFIRM prompt with
        THEN RETURN;                                   !  CTRL/Z at some point don't delete any more files.
!
! If the /LOG bit is set or OPEN_FILE bit is set, then open the current file to get
! the block size of the file.  An error
! code of RMS-FNF at this point means that we have a dangling directory
! entry "no such file", and we want to delete these if possible.
!

    IF .del$cli_status [del$v_log_msg] OR              ! If /LOG qualifier was given
       .del$cli_status [del$v_open_file]               ! or OPEN_FILE bit is set, then open file.
    THEN
        BEGIN
        fab_block [fab$l_alq] = 0;                     ! Init block size of file to 0 in case $OPEN fails.
        $OPEN (FAB = .fab_block);                      ! Open the file.
        del$file_size = .fab_block [fab$l_alq];        ! Get block size of file (0 if $OPEN failed).
        END;

!
! If the /CONFIRM qualifier was specified, ask the user whether this file
! should be deleted. If the user says no, then just go process the next file.
! Also, if user answers CONTROL/Z stop all processing.
!

    name_desc[0] = .nam_block[nam$b_rsl];              ! Make descriptor pointing to related
    name_desc[1] = .nam_block[nam$l_rsa];              !  file name in case /CONFIRM requested.
    prompt_desc[0] = name_desc;
    status = lib$qual_file_match( del$context,         ! Context pointer.
                                  .fab_block,          ! Fab pointer.
                                  0,                   ! No file name.
                                  $descriptor('!AS, delete? [N]:'),      ! Prompt string.
                                  prompt_desc,         ! Prompt arguments.
                                  0);                  ! No prompt routine.

    IF NOT .status
    THEN
        BEGIN
        IF .status EQL lib$_quipro                ! If user said CNTRL/Z
        THEN                                      !  then stop processing.
            del$cli_status [del$v_cntrl_z_stop] = TRUE;
        IF (.status NEQ lib$_quipro) AND          ! If user said CNTRL/Z
           (.status NEQ lib$_filfaimat)           !  or file did not meet criteria
        THEN                                      !  then do not report an error.
            del$file_error(msg$_filnotacc,.fab_block);
        $CLOSE ( FAB = .fab_block);               ! Ask RMS to close the file.
        RETURN;
        END;

!
! Delete the current file. If /ERASE is requested, mark the file for erase first.
! If the file is open because the /LOG qualifier was requested, just close it
! with the delete bit set. Otherwise simply call RMS to perform the erase function.
! If the erase or delete doesn't work, just return. The calling routine
! will go on to process the next file.
```

```
     2 !
     2
     2      IF .del$cli_status [del$v_erase]                       ! If erase requested
     2      THEN
     3          BEGIN
     3          $CLOSE (FAB = .fab_block);                         ! close the file so we can do it.
     3          status = lib$set_erase (name_desc);                ! do it.
     3          fab_block[fab$l_sts] = .status;
     3          fab_block[fab$l_stv] = 0;
     2          END;
     2
     2      IF .status
     2      THEN
     3          BEGIN
     3          IF .fab_block [fab$w_ifi] NEQ 0                    ! If the file is open,
     3          THEN
     4              BEGIN
     4              fab_block [fab$v_dlt] = TRUE;                  !  then set the deletion bit,
     4              status = $CLOSE ( FAB = .fab_block);           !  and ask RMS to close and delete the file.
     4              fab_block [fab$v_dlt] = FALSE;                 ! Turn off the delete bit to avoide side effects.
     4              END
     3          ELSE
     3              status = $ERASE ( FAB = .fab_block);           ! Erase the file.
     2          END;
     2
     2      IF NOT .status                                         ! If the ERASE function fails
     2      THEN
     3          BEGIN
     3          del$file_error ( msg$_filnotdel, .fab_block);      !  then report an error to the user,
     3          $CLOSE (FAB=.fab_block);                           !  close the file quietly
     3          RETURN;                                            !  and return.
     2          END;
     2
     2 !
     2 ! If the /LOG qualifier was given, report the deletion.
     2 !
     2
     2      IF .del$cli_status [del$v_log_msg]
     2      THEN                                                   ! If successful deletion and logging going on,
     3          BEGIN                                              !    then report the deletion.
     3          del$blocks_deleted = .del$blocks_deleted + .del$file_size; ! Keep running total of blocks deleted
     3          del$files_deleted = .del$files_deleted + 1;        ! Increment "number of files deleted" counter
   P 3          put_message (                                      ! and output the message.
   P 3                          msg$_fildel,                       !    This is the message number,
   P 3                          2,                                 !    two arguments coming, which are
   P 3                          name_desc,                         !    the file name descriptor and
     3                          .del$file_size);                   !  the block size of the file.
     2          END;
     2
     1      END;


                                                     .PSECT  $PLIT$,NOWRT,NOEXE,2

4E 5B 20 3F 65 74 65 6C 65 64 20 2C 53 41 21  000C0 P.AAG:  .ASCII  \!AS, delete? [N]:\          :
                                       3A 5D  000CF                                             :
                                            000D1            .BLKB    3
                                 00000011   000D4 P.AAF:     .LONG    17                                      ;
                                 00000000'  000D8            .ADDRESS P.AAG                                   ;

                                                             .EXTRN   SYS$OPEN, SYS$CLOSE
                                                             .EXTRN   SYS$ERASE

                                                             .PSECT   $CODE$,NOWRT,2

                                       007C 00000 DELETE_FILE:
                                                             .WORD    Save R2,R3,R4,R5,R6                     ; 0382
              56 00000000G  00  9E 00002                     MOVAB    LIB$ QUIPRO, R6                         ;
              55 00000000G  00  9E 00009                     MOVAB    SYS$CLOSE, R5                           ;
              54      0000' CF  9E 00010                     MOVAB    DEL$CLI_STATUS, R4                      ;
              5E            10  C2 00015                     SUBL2    #16, SP                                 ;
              52        04  AC  D0 00018                     MOVL     FAB BLOCK, R2                           ; 0434
              53        28  A2  D0 0001C                     MOVL     40(R2), R3                              ;
   01         64            05  E1 00020                     BBC      #5, DEL$CLI_STATUS, 1$                  ; 0442
                                04 00024                     RET                                              ;
   04         64            01  E0 00025 1$:                 BBS      #1, DEL$CLI_STATUS, 2$                  ; 0451
                            64  95 00029                     TSTB     DEL$CLI_STATUS                          ; 0452
                            11  18 0002B                     BGEQ     3$                                      ;
                        10  A2  D4 0002D 2$:                 CLRL     16(R2)                                  ; 0455
                            52  DD 00030                     PUSHL    R2                                      ; 0456
      00000000G   00        01  FB 00032                     CALLS    #1, SYS$OPEN                            ;
             10   A4    10  A2  D0 00039                     MOVL     16(R2), DEL$FILE_SIZE                   ; 0457
                  6E    03  A3  9A 0003E 3$:                 MOVZBL   3(R3), NAME_DESC                        ; 0466
             04   AE    04  A3  D0 00042                     MOVL     4(R3), NAME_DESC+4                      ; 0467
             08   AE        6E  9E 00047                     MOVAB    NAME DESC, PROMPT_DESC                  ; 0468
                            7E  D4 0004B                     CLRL     -(SP)                                   ; 0469
                        0C  AE  9F 0004D                     PUSHAB   PROMPT_DESC                             ;
                     0000'  CF  9F 00050                     PUSHAB   P.AAF                                   ; 0472
                            7E  D4 00054                     CLRL     -(SP)                                   ; 0469
                            52  DD 00056                     PUSHL    R2                                      ; 04:0
                        FC  A4  9F 00058                     PUSHAB   DEL$CONTEXT                             ; 0469
      00000000G   00        06  FB 0005B                     CALLS    #6, LIB$QUAL_FILE_MATCH                 ;
                  53        50  D0 00062                     MOVL     R0, STATUS                              ;
                  29        53  E8 00065                     BLBS     STATUS, 5$                              ; 0476
                  50        66  9E 00068                     MOVAB    LIB$ QUIPRO, R0                         ; 0479
                  50        53  D1 0006B                     CMPL     STATUS, R0                              ;
                            03  12 0006E                     BNEQ     4$                                      ;
                  64        20  88 00070                     BISB2    #32, DEL$CLI_STATUS                     ; 0481
                  50        66  9E 00073 4$:                 MOVAB    LIB$ QUIPRO, R0                         ; 0482
                  50        53  D1 00076                     CMPL     STATUS, R0                              ;
                            6A  13 00079                     BEQL     11$                                     ;
                  50 00000000G  00  9E 0007B                 MOVAB    LIB$ FILFAIMAT, R0                      ; 0483
                  50        53  D1 00082                     CMPL     STATUS, R0                              ;
                            5E  13 00085                     BEQL     11$                                     ;
                            52  DD 00087                     PUSHL    R2                                      ; 0485
                  00931338  8F  DD 00089                     PUSHL    #9638712                                ;
                            4F  11 0008F                     BRB      10$                                     ;
   18             64        04  E1 00091 5$:                 BBC      #4, DEL$CLI_STATUS, 6$                  ; 0498
                            52  DD 00095                     PUSHL    R2                                      ; 0501
                  65        01  FB 00097                     CALLS    #1, SYS$CLOSE                           ;
                            5E  DD 0009A                     PUSHL    SP                                      ; 0502
      00000000G   00        01  FB 0009C                     CALLS    #1, LIB$SET_ERASE                       ;
                  53        50  D0 000A3                     MOVL     R0, STATUS                              ;
```

```
               08  A2             53  D0 000A6          MOVL    STATUS, 8(R2)                               ; 0503
                              0C  A2  D4 000AA          CLRL    12(R2)                                      ; 0504
                   28             53  E9 000AD 6$:      BLBC    STATUS, 9$                                  ; 0507
                              02  A2  B5 000B0          TSTW    2(R2)                                       ; 0510
                                  14  13 000B3          BEQL    7$                                          ;
               05  A2         80  8F  88 000B5          BISB2   #128, 5(R2)                                 ; 0513
                                  52  DD 000BA          PUSHL   R2                                          ; 0514
                   65             01  FB 000BC          CALLS   #1, SYS$CLOSE                               ;
                   53             50  D0 000BF          MOVL    R0, STATUS                                  ;
               05  A2         80  8F  BA 000C2          BICB2   #128, 5(R2)                                 ; 0515
                                  0C  11 000C7          BRB     8$                                          ; 0510
                                  52  DD 000C9 7$:      PUSHL   R2                                          ; 0518
        00000000G  00             01  FB 000CB          CALLS   #1, SYS$ERASE                               ;
                   53             50  D0 000D2          MOVL    R0, STATUS                                  ;
                   13             53  E8 000D5 8$:      BLBS    STATUS, 12$                                 ; 0521
                                  52  DD 000D8 9$:      PUSHL   R2                                          ; 0524
                        009311F0  8F  DD 000DA          PUSHL   #9638384                                    ;
            0000V  CF             02  FB 000E0 10$:     CALLS   #2, DEL$FILE_ERROR                          ;
                                  52  DD 000E5 11$:     PUSHL   R2                                          ; 0525
                   65             01  FB 000E7          CALLS   #1, SYS$CLOSE                               ;
                                      04 000EA          RET                                                 ; 0523
  1D               64             01  E1 000EB 12$:     BBC     #1, DEL$CLI_STATUS, 13$                     ; 0533
               0C  A4         10  A4  C0 000EF          ADDL2   DEL$FILE_SIZE, DEL$BLOCKS_DELETED           ; 0536
                              08  A4  D6 000F4          INCL    DEL$FILES_DELETED                           ; 0537
                              10  A4  DD 000F7          PUSHL   DEL$FILE_SIZE                               ; 0542
                              04  AE  9F 000FA          PUSHAB  NAME_DESC                                   ;
                                  02  DD 000FD          PUSHL   #2                                          ;
                        00931323  8F  DD 000FF          PUSHL   #9638691                                    ;
        00000000G  00             04  FB 00105          CALLS   #4, LIB$SIGNAL                              ;
                                      04 0010C 13$:     RET                                                 ; 0545

; Routine Size:  269 bytes,    Routine Base:  $CODE$ + 00FC


;  444          0546  1
```

```
1 ROUTINE del$get_cmdqual :  NOVALUE =
1
1 !++
1 ! Functional description:
1 !
1 !       This routine calls the CLI to obtain the command line. Then the routine
1 !       obtains the command-level qualifiers (i.e., options) from the
1 !       Command Language Interpreter.
1 !
1 ! Calling sequence:
1 !
1 !       del$get_cmdqual ()
1 !
1 ! Input parameters
1 !
1 !       none
1 !
1 ! Output parameters
1 !
1 !       none
1 !
1 ! Implicit outputs
1 !
1 !       DEL$CLI_STATUS  - Relevant command and qualifier indicators set by CLI.
1 !                         PURGE_CMD flag set if this was a PURGE command.
1 !
1 ! Routine value
1 !
1 !       novalue
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2     bitmap,                                               ! Contains map of qualifiers requested
2     status,
2     char;                                                 ! Holds one character
2
2 CH$FILL(0, dsc$c_s_bln, infile_desc);                    ! Make descriptor dynamic
2 infile_desc[dsc$b_class] = dsc$k_class_d;
2
2 !
2 ! Determine whether this is a DELETE or a PURGE command and get qualifers.
2 !
2
2 cli$get_value($descriptor('$VERB'),infile_desc);        ! Get command
3 IF (CH$RCHAR(.infile_desc [dsc$a_pointer]) EQL 'P')     ! If first letter is P
2 THEN
3     BEGIN
3     del$cli_status[del$v_purge_cmd] = true;              ! Note that this is a PURGE command
3     del$cli_status[del$v_keep] =        cli$present($descriptor('KEEP'));
2     END;
2
2 del$cli_status[del$v_erase] =           cli$present($descriptor('ERASE'));
2 del$cli_status[del$v_log_msg] =         cli$present($descriptor('LOG'));
2 del$cli_status[del$v_conf_prompt] =     cli$present($descriptor('CONFIRM'));
```

```
2
2
2  IF .del$cli_status [del$v_keep]                 ! /KEEP present, so get its value
2  THEN
3      BEGIN
3      cli$get_value($descriptor('KEEP'), infile_desc);
3      status = lib$cvt_dtb(.infile_desc [dsc$w_length],   ! Move value into Global variable
3              .infile_desc [dsc$a_pointer], del$keepver_val);
4      IF NOT .status OR (.del$keepver_val LEQ 0)  ! If value is LEQ 0 then
3      THEN                                        !   report an error.
3          SIGNAL (msg$_badvalue,1,infile_desc);
2      END;
2
2
2  bitmap= lib$m_cqf_confirm  OR  lib$m_cqf_exclude  OR    ! Bitmap of qualifiers
2          lib$m_cqf_before   OR  lib$m_cqf_since    OR    !  that the common qualifier
2          lib$m_cqf_created  OR  lib$m_cqf_modified OR    !  package is to check for.
2          lib$m_cqf_expired  OR  lib$m_cqf_backup   OR
2          lib$m_cqf_byowner;
2
3  IF NOT (status = lib$qual_file_parse(bitmap,del$context))       ! Get common qualifiers
2  THEN
2      SIGNAL_STOP(.status);
2
2  !
2  ! If any of the following qualifers were specified, then always open the file
2  ! when we process it.  This is an optimization to cut down on the number of
2  ! $OPEN's and $CLOSE's that must be done to delete or purge a file.
2  !
2  IF  cli$present($descriptor('BEFORE')) OR cli$present($descriptor('EXCLUDE'))
2   OR cli$present($descriptor('SINCE'))  OR cli$present($descriptor('BY_OWNER'))
2  THEN del$cli_status[del$v_open_file] = TRUE;
2
1  END;


                                                      .PSECT  $PLIT$,NOWRT,NOEXE,2

             42  52  45  56  24  000DC P.AAI:   .ASCII  \$VERB\                          ;
                                 000E1          .BLKB   3
                      00000005   000E4 P.AAH:   .LONG   5                                ;
                      00000000'  000E8          .ADDRESS P.AAI
                 50  45  45  4B  000EC P.AAK:   .ASCII  \KEEP\
                      00000004   000F0 P.AAJ:   .LONG   4
                      00000000'  000F4          .ADDRESS P.AAK
             45  53  41  52  45  000F8 P.AAM:   .ASCII  \ERASE\
                                 000FD          .BLKB   3
                      00000005   00100 P.AAL:   .LONG   5                                ;
                      00000000'  00104          .ADDRESS P.AAM
                     47  4F  4C  00108 P.AAO:   .ASCII  \LOG\
                                 0010B          .BLKB   1
                      00000003   0010C P.AAN:   .LONG   3                                ;
                      00000000'  00110          .ADDRESS P.AAO
     4D  52  49  46  4E  4F  43  00114 P.AAQ:   .ASCII  \CONFIRM\
                                 0011B          .BLKB   1
                      00000007   0011C P.AAP:   .LONG   7                                ;
```

```
                                      00000000'  00120            .ADDRESS P.AAQ                                 ;
                        50   45   45   4B  00124  P.AAS:   .ASCII   \KEEP\                                       ;
                                      00000004   00128  P.AAR:   .LONG    4                                      ;
                                      00000000'  0012C            .ADDRESS P.AAS                                 ;
              45   52   4F   46   45   42  00130  P.AAU:   .ASCII   \BEFORE\                                     ;
                                                 00136            .BLKB    2
                                      00000006   00138  P.AAT:   .LONG    6                                      ;
                                      00000000'  0013C            .ADDRESS P.AAU                                 ;
         45   44   55   4C   43   58   45  00140  P.AAW:   .ASCII   \EXCLUDE\                                    ;
                                                 00147            .BLKB    1
                                      00000007   00148  P.AAV:   .LONG    7                                      ;
                                      00000000'  0014C            .ADDRESS P.AAW                                 ;
                   45   43   4E   49   53  00150  P.AAY:   .ASCII   \SINCE\                                      ;
                                                 00155            .BLKB    3
                                      00000005   00158  P.AAX:   .LONG    5                                      ;
                                      00000000'  0015C            .ADDRESS P.AAY                                 ;
    52   45   4E   57   4F   5F   59   42  00160  P.ABA:   .ASCII   \BY_OWNER\                                   ;
                                      00000008   00168  P.AAZ:   .LONG    8                                      ;
                                      00000000'  0016C            .ADDRESS P.ABA                                 ;



                                                                  .PSECT  $CODE$,NOWRT,2

                                           03FC  00000  DEL$GET_CMDQUAL:
                                                                  .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9          ; 0547
                    59 00000000G   00   9E  00002            MOVAB    CLI$GET_VALUE, R9                             ;
                    58     0000'   CF   9E  00009            MOVAB    P.AAH, R8                                     ;
                    57 00000000G   00   9E  0000E            MOVAB    CLI$PRESENT, R7                               ;
                    56     0000'   CF   9E  00015            MOVAB    DEL$CLI_STATUS, R6                            ;
                    5E             04   C2  0001A            SUBL2    #4, SP                                        ;
   08          00   6E             00   2C  0001D            MOVC5    #0, (SP), #0, #8, INFILE_DESC                 ; 0586
                            18     A6       00022                                                                   ;
             1B   A6               02   90  00024            MOVB     #2, INFILE_DESC+3                             ; 0587
                            18     A6   9F  00028            PUSHAB   INFILE_DESC                                   ; 0593
                                   58   DD  0002B            PUSHL    R8                                            ;
                    69             02   FB  0002D            CALLS    #2, CLI$GET_VALUE                             ;
             50   8F        1C     B6   91  00030            CMPB     @INFILE_DESC+4, #80                           ; 0594
                                   0E   12  00035            BNEQ     1$                                            ;
                    66             08   88  00037            BISB2    #8, DEL$CLI_STATUS                            ; 0597
                            0C     A8   9F  0003A            PUSHAB   P.AAJ                                         ; 0598
                    67             01   FB  0003D            CALLS    #1, CLI$PRESENT                               ;
   66          01   02             50   F0  00040            INSV     R0, #2, #1, DEL$CLI_STATUS                    ;
                            1C     A8   9F  00045  1$:       PUSHAB   P.AAL                                         ; 0601
                    67             01   FB  00048            CALLS    #1, CLI$PRESENT                               ;
   66          01   04             50   F0  0004B            INSV     R0, #4, #1, DEL$CLI_STATUS                    ;
                            28     A8   9F  00050            PUSHAB   P.AAN                                         ; 0602
                    67             01   FB  00053            CALLS    #1, CLI$PRESENT                               ;
   66          01   01             50   F0  00056            INSV     R0, #1, #1, DEL$CLI_STATUS                    ;
                            38     A8   9F  0005B            PUSHAB   P.AAP                                         ; 0603
                    67             01   FB  0005E            CALLS    #1, CLI$PRESENT                               ;
   66          01   06             50   F0  00061            INSV     R0, #6, #1, DEL$CLI_STATUS                    ;
               37   66             02   E1  00066            BBC      #2, DEL$CLI_STATUS, 3$                        ; 0606
                            18     A6   9F  0006A            PUSHAB   INFILE_DESC                                   ; 0609
                            44     A8   9F  0006D            PUSHAB   P.AAR                                         ;
                    69             02   FB  00070            CALLS    #2, CLI$GET_VALUE                             ;
                            04     A6   9F  00073            PUSHAB   DEL$KEEPVER_VAL                               ; 0610
```

```
                                     1C   A6  DD 00076          PUSHL    INFILE_DESC+4                        ; 0611
                              7E     18   A6  3C 00079          MOVZWL   INFILE_DESC, -(SP)                   ; 0610
                   00000000G  00          03  FB 0007D          CALLS    #3, LIB$CVT_DTB                      ;
                              52          50  D0 00084          MOVL     R0, STATUS                           ;
                              05          52  E9 00087          BLBC     STATUS, 2$                           ; 0612
                                     04   A6  D5 0008A          TSTL     DEL$KEEPVER_VAL                      ;
                                          12  14 0008D          BGTR     3$                                   ;
                                     18   A6  9F 0008F  2$:     PUSHAB   INFILE_DESC                          ; 0614
                                          01  DD 00092          PUSHL    #1                                   ;
                               00931114   8F  DD 00094          PUSHL    #9638164                             ;
                   00000000G  00          03  FB 0009A          CALLS    #3, LIB$SIGNAL                       ;
                              6E   01FF   8F  3C 000A1  3$:     MOVZWL   #511, BITMAP                         ; 0621
                                     FC   A6  9F 000A6          PUSHAB   DEL$CONTEXT                          ; 0624
                                     04   AE  9F 000A9          PUSHAB   BITMAP                               ;
                   00000000G  00          02  FB 000AC          CALLS    #2, LIB$QUAL_FILE_PARSE              ;
                              52          50  D0 000B3          MOVL     R0, STATUS                           ;
                              09          52  E8 000B6          BLBS     STATUS, 4$                           ;
                                          52  DD 000B9          PUSHL    STATUS                               ; 0626
                   00000000G  00          01  FB 000BB          CALLS    #1, LIB$STOP                         ;
                                     54   A8  9F 000C2  4$:     PUSHAB   P.AAT                                ; 0633
                              67          01  FB 000C5          CALLS    #1, CLI$PRESENT                      ;
                              1C          50  E8 000C8          BLBS     R0, 5$                               ;
                                     64   A8  9F 000CB          PUSHAB   P.AAV                                ;
                              67          01  FB 000CE          CALLS    #1, CLI$PRESENT                      ;
                              13          50  E8 000D1          BLBS     R0, 5$                               ;
                                     74   A8  9F 000D4          PUSHAB   P.AAX                                ; 0634
                              67          01  FB 000D7          CALLS    #1, CLI$PRESENT                      ;
                              0A          50  E8 000DA          BLBS     R0, 5$                               ;
                                   0084   C8  9F 000DD          PUSHAB   P.AAZ                                ;
                              67          01  FB 000E1          CALLS    #1, CLI$PRESENT                      ;
                              04          50  E9 000E4          BLBC     R0, 6$                               ;
                              66     80   8F  88 000E7  5$:     BISB2    #128, DEL$CLI_STATUS                 ; 0635
                                              04 000EB  6$:     RET                                           ; 0637

; Routine Size:  236 bytes,    Routine Base:  $CODE$ + 0209

;  537          0638  1
```

```
  1 ROUTINE del$get_file (fab_block, nam_block, xab_chain) =
  1
  1 !++
  1 ! Functional description:
  1 !
  1 !       This routine gets an file specification and all
  1 !       related qualifiers from the Command Language Interpreter. Then
  1 !       the file specification is parsed.
  1 !
  1 !       If no more input specifications are available, this routine just
  1 !       returns successfully.
  1 !
  1 ! Input parameters:
  1 !
  1 !       fab_block       - the FAB to use for this specification
  1 !       nam_block       - the NAM to use for this specification
  1 !       xab_chain       - the appropriate XAB block's
  1 !
  1 ! Implicit inputs:
  1 !
  1 !       infile_desc     - CLI block for file specifications
  1 !
  1 ! Output parameters:
  1 !
  1 !       none
  1 !
  1 ! Implicit outputs:
  1 !
  1 !       The fields of the FAB and the NAM block are filled in according
  1 !       to the CLI call and the $PARSE function call.
  1 !
  1 ! Routine value:
  1 !
  1 !       TRUE            - success
  1 !       NO_MORE_FILES   - success, no more file specifications
  1 !       NO_FILE         - failure
  1 !
  1 !--
  1
  2     BEGIN
  2
  2     MAP
  2         fab_block       : REF $BBLOCK,                  ! FAB to use with file
  2         nam_block       : REF $BBLOCK;                  ! NAM to use with file
  2
  2     LOCAL
  2         status;
  2
  2 !
  2 ! Get a file specification from the Command Language Interpreter.
  2 !
  2
  2     IF NOT cli$get_value($descriptor('INPUT'), infile_desc)     ! If no more file specs are coming,
  2     THEN                                                        !  then return successfully, without a file.
  2         RETURN no_more_files;
  2
P 2     $FAB_INIT (                                                 ! Setup the file FAB as follows:
```

```
;   596      P 0696  2                  FAB = .fab_block,                  !     FAB address is the input parameter.
;   597      P 0697  2                  FAC = <GET>,                       !     Input file.
;   598      P 0698  2                  DNS = 0,                           !     No default file specification.
;   599      P 0699  2                  FNA = .infile_desc [dsc$a_pointer], !    Move the file name address
;   600      P 0700  2                  FNS = .infile_desc [dsc$w_length], !      and length into the file FAB block.
;   601      P 0701  2                  FOP = <NAM>,                       !     Open by name block.
;   602      P 0702  2                  NAM = .nam_block,                  !     NAM block address.
;   603        0703  2                  XAB = .xab_chain);                 !     XAB block address.
;   604        0704  2
;   605        0705  2   !
;   606        0706  2   ! Call RMS $PARSE function to parse the file specification. This resolves logical names and
;   607        0707  2   ! determines if there are wildcards present, or explicit named fields present.
;   608        0708  2   !
;   609        0709  2
;   610        0710  2       nam_block[nam$v_synchk] = true;
;   611        0711  2       status = $PARSE ( FAB = .fab_block);                 ! Call the RMS file-spec parsing routine,
;   612        0712  2       nam_block[nam$v_synchk] = false;
;   613        0713  2       IF NOT .status
;   614        0714  2       THEN
;   615        0715  3           BEGIN                                         ! If unsuccessful then
;   616        0716  3           del$file_error (msg$_filnotdel, .fab_block);  !  signal an error and
;   617        0717  3           RETURN no_file;                               ! Return an error to the caller.
;   618        0718  2           END;
;   619        0719  2
;   620        0720  2
;   621        0721  2   !
;   622        0722  2   ! Don't allow version number specification for a PURGE command.  If the file
;   623        0723  2   ! name and file type were not specified on the PURGE command, default to
;   624        0724  2   ! "*.*".  In any case, always add a version ';*'.  A DELETE command,
;   625        0725  2   ! to the contrary, requires an explicit or wildcard version number.
;   626        0726  2   !
;   627        0727  2
;   628        0728  2       IF .del$cli_status [del$v_purge_cmd]                 ! If this is a PURGE command,
;   629        0729  2       THEN
;   630        0730  3           BEGIN
;   631        0731  3           IF .nam_block [nam$v_wild_ver] OR                 !   then look for version number
;   632        0732  3               .nam_block [nam$v_exp_ver]                    !   specification.
;   633        0733  3           THEN
;   634        0734  4               BEGIN
;   635        0735  4               put_message (msg$_purgever);                  ! If present, signal an error,
;   636        0736  4               RETURN bad_version;                           !   and return an error status code.
;   637        0737  3               END;
;   638        0738  3
;   639        0739  3           IF  NOT .nam_block[nam$v_exp_name]                ! If file name missing
;   640        0740  3           AND NOT .nam_block[nam$v_wild_name]
;   641        0741  3           AND NOT .nam_block[nam$v_exp_type]                ! and file type missing
;   642        0742  3           AND NOT .nam_block[nam$v_wild_type]
;   643        0743  3           THEN
;   644        0744  4               BEGIN                                         ! THEN
;   645        0745  4               fab_block[fab$l_dna] =                        ! Substitute defaults
;   646        0746  4                   UPLIT BYTE (%ASCII '*.*;*');
;   647        0747  4               fab_block[fab$b_dns] = 5;
;   648        0748  4               END
;   649        0749  3           ELSE
;   650        0750  4               BEGIN
;   651        0751  4               fab_block [fab$l_dna] = UPLIT(';*');         ! Set default name string to ';*'
;   652        0752  4               fab_block [fab$b_dns] = 2;
;  653          0753  3                 END;
;  654          0754  3             END
;  655          0755  2         ELSE
;  656          0756  3             BEGIN                                       ! However, is this is a DELETE command
;  657          0757  3             IF   NOT  .nam_block [nam$v_wild_ver] AND   !   and an explicit or wildcard version
;  658          0758  3                  NOT  .nam_block [nam$v_exp_ver] AND    !   number is not present,
;  659          0759  4                  NOT (.nam_block [nam$v_quoted] AND     !   and the file specification
;  660          0760  4                       .nam_block [nam$v_node])          !   is not a network quoted filespec
;  661          0761  3             THEN
;  662          0762  4                 BEGIN
;  663          0763  4                 put_message (msg$_delver);              !   then signal an error,
;  664          0764  4                 RETURN bad_version;                     !   and return an error status code.
;  665          0765  3                 END;
;  666          0766  2             END;
;  667          0767  2
;  668          0768  2         RETURN true;                                    ! Return a success code
;  669          0769  1         END;


                                                              .PSECT  $PLIT$,NOWRT,NOEXE,2

                         54  55  50  4E  49  00170 P.ABC:     .ASCII  \INPUT\                                          ;
                                             00175            .BLKB   3
                                   00000005  00178 P.ABB:     .LONG   5                                                ;
                                   00000000' 0017C            .ADDRESS P.ABC                                           ;
                         2A  3B  2A  2E  2A  00180 P.ABD:     .ASCII  \*.*;*\                                          ;
                                             00185            .BLKB   3                                                ;
                             00  00  2A  3B  00188 P.ABE:     .ASCII  \;*\<0><0>                                       ;

                                                              .EXTRN  SYS$PARSE

                                                              .PSECT  $CODE$,NOWRT,2

                                        00FC 00000 DEL$GET_FILE:
                                                              .WORD   Save R2,R3,R4,R5,R6,R7                           ; 0639
                         57      0000'  CF  9E 00002           MOVAB   INFILE_DESC, R7                                  ;
                                        57  DD 00007           PUSHL   R7                                               ; 0691
                                 0000'  CF  9F 00009           PUSHAB  P.ABB                                            ;
                 00000000G  00          02  FB 0000D           CALLS   #2, CLI$GET_VALUE                                ;
                            04          50  E8 00014           BLBS    R0, 1$                                           ;
                            50          03  D0 00017           MOVL    #3, R0                                           ; 0693
                                            04 0001A           RET                                                      ;
                            56     04   AC  D0 0001B 1$:       MOVL    FAB_BLOCK, R6                                    ; 0703
  0050   8F         00      6E          00  2C 0001F           MOVC5   #0, (SP), #0, #80, (R6)                          ;
                                        66     00026                                                                    ;
                            66   5003   8F  B0 00027           MOVW    #20483, (R6)                                     ;
                    04  A6  01000000    8F  D0 0002C           MOVL    #16777216, 4(R6)                                 ;
                    16  A6              02  90 00034           MOVB    #2, 22(R6)                                       ;
                    1F  A6              02  90 00038           MOVB    #2, 31(R6)                                       ;
                    24  A6         0C   AC  D0 0003C           MOVL    XAB_CHAIN, 36(R6)                                ;
                        52         08   AC  D0 00041           MOVL    NAM_BLOCK, R2                                    ;
                    28  A6              52  D0 00045           MOVL    R2, 40(R6)                                       ;
                    2C  A6         04   A7  D0 00049           MOVL    INFILE_DESC+4, 44(R6)                            ;
                    34  A6              67  90 0004E           MOVB    INFILE_DESC, 52(R6)                              ;
                    08  A2              08  88 00052           BISB2   #8, 8(R2)                                        ; 0710
                                        56  DD 00056           PUSHL   R6                                               ; 0711
```

```
                               000000000G  00              01  FB 00058           CALLS    #1, SYS$PARSE               ;
                                       08  A2              08  8A 0005F           BICB2    #8, 8(R2)                   ; 0712
                                           0F              50  E8 00063           BLBS     STATUS, 2$                  ; 0713
                                                           56  DD 00066           PUSHL    R6                          ; 0716
                                               009311F0    8F  DD 00068           PUSHL    #9638384                    ;
                                    0000V  CF              02  FB 0006E           CALLS    #2, DEL$FILE_ERROR          ;
                                                           63  11 00073           BRB      10$                         ; 0717
                                           52              34  C0 00075 2$:       ADDL2    #52, R2                     ; 0731
                            37         E8  A7              03  E1 00078           BBC      #3, DEL$CLI_STATUS, 6$      ; 0728
                            03             62              03  E0 0007D           BBS      #3, (R2), 3$                ; 0731
                                           08              62  E9 00081           BLBC     (R2), 4$                    ; 0732
                                               00931212    8F  DD 00084 3$:       PUSHL    #9638418                    ; 0735
                                                           3D  11 0008A           BRB      8$                          ;
                            18             62              02  E0 0008C 4$:       BBS      #2, (R2), 5$                ; 0739
                            14             62              05  E0 00090           BBS      #5, (R2), 5$                ; 0740
                            10             62              01  E0 00094           BBS      #1, (R2), 5$                ; 0741
                            0C             62              04  E0 00098           BBS      #4, (R2), 5$                ; 0742
                                       30  A6     0000'    CF  9E 0009C           MOVAB    P.ABD, 48(R6)               ; 0746
                                       35  A6              05  90 000A2           MOVB     #5, 53(R6)                  ; 0747
                                                           2C  11 000A6           BRB      9$                          ; 0739
                                       30  A6     0000'    CF  9E 000A8 5$:       MOVAB    P.ABE, 48(R6)               ; 0751
                                       35  A6              02  90 000AE           MOVB     #2, 53(R6)                  ; 0752
                                                           20  11 000B2           BRB      9$                          ; 0728
                            1C             62              03  E0 000B4 6$:       BBS      #3, (R2), 9$                ; 0757
                                           19              62  E8 000B8           BLBS     (R2), 9$                    ; 0758
                            04             62              12  E1 000BB           BBC      #18, (R2), 7$               ; 0759
                            11             62              11  E0 000BF           BBS      #17, (R2), 9$               ; 0760
                                               0093120A    8F  DD 000C3 7$:       PUSHL    #9638410                    ; 0763
                               000000000G  00              01  FB 000C9 8$:       CALLS    #1, LIB$SIGNAL              ;
                                           50              02  D0 000D0           MOVL     #2, R0                      ; 0764
                                                               04 000D3           RET                                  ;
                                           50              01  D0 000D4 9$:       MOVL     #1, R0                      ; 0768
                                                               04 000D7           RET                                  ;
                                                           50  D4 000D8 10$:      CLRL     R0                          ; 0769
                                                               04 000DA           RET                                  ;

; Routine Size:  219 bytes,    Routine Base:  $CODE$ + 02F5


;  670           0770  1
;  671           0771  1
;  673          0772  1 GLOBAL ROUTINE del$search_error (fab_block) : NOVALUE =
;  674          0773  1
;  675          0774  1 !++
;  676          0775  1 ! Functional description:
;  677          0776  1 !
;  678          0777  1 !       This routine reports an error as a result of searching for the
;  679          0778  1 !       next file to be deleted.
;  680          0779  1 !
;  681          0780  1 ! Calling sequence:
;  682          0781  1 !
;  683          0782  1 !       delete_error (fab_block.ra.v)
;  684          0783  1 !
;  685          0784  1 ! Input parameters:
;  686          0785  1 !
;  687          0786  1 !       fab_block       - the FAB associated with the file
;  688          0787  1 !
;  689          0788  1 ! Implicit inputs:
;  690          0789  1 !
;  691          0790  1 !       none
;  692          0791  1 !
;  693          0792  1 ! Output parameters:
;  694          0793  1 !
;  695          0794  1 !       none
;  696          0795  1 !
;  697          0796  1 ! Implicit outputs:
;  698          0797  1 !
;  699          0798  1 !       none
;  700          0799  1 !
;  701          0800  1 ! Routine value:
;  702          0801  1 !
;  703          0802  1 !       none
;  704          0803  1 !
;  705          0804  1 ! Side effects:
;  706          0805  1 !
;  707          0806  1 !       none
;  708          0807  1 !
;  709          0808  1 !--
;  710          0809  1
;  711          0810  2     BEGIN
;  712          0811  2
;  713          0812  2     del$file_error(msg$_searchfail,.fab_block);        ! Report specified RMS error.
;  714          0813  2
;  715          0814  1     END;



                                      0000 00000        .ENTRY  DEL$SEARCH_ERROR, Save nothing                     ; 0772
                                04  AC  DD 00002        PUSHL   FAB_BLOCK                                          ; 0812
                          00931238  8F  DD 00005        PUSHL   #9638456                                           ;
                  0000V  CF         02  FB 0000B        CALLS   #2, DEL$FILE_ERROR                                 ;
                                        04 00010        RET                                                        ; 0814

; Routine Size:  17 bytes,    Routine Base:  $CODE$ + 03D0
;  716          0815  1
```

```
1 GLOBAL ROUTINE del$file_error (message_id, fab_block) : NOVALUE =
1
1 !++
1 ! Functional description
1 !
1 !       This RMS error action routine sends an error message to the user.
1 !
1 ! Calling sequence
1 !
1 !       del$file_error (message_id.rv, fab_block.ra.v)
1 !
1 ! Input parameters
1 !
1 !       message_id      - The message code for the message to send.
1 !       fab_block       - Address of the FAB block of the file for which the error occurred
1 !
1 ! Implicit inputs
1 !
1 !       The associated NAM block.
1 !
1 !       INFILE_DESC     - the CLI data block for the parameter
1 !
1 ! Output parameters
1 !
1 !       none
1 !
1 ! Implicit outputs
1 !
1 !       none
1 !
1 ! Routine value
1 !
1 !       novalue
1 !
1 ! Side effects
1 !
1 !       none
1 !
1 !--
1
2    BEGIN
2
2    MAP
2       fab_block       : REF $BBLOCK;
2
2    BIND
2       nam_block = .fab_block [fab$l_nam]  : $BBLOCK;  ! Associated NAM block address
2
2    LOCAL
2       name_desc       : VECTOR [2];                   ! String descriptor for the file name
2
2 !
2 ! Fill in the file name descriptor with the most complete name possible.
2 !
2
2    IF .nam_block [nam$b_rsl] NEQ 0                    ! If a resultant name string exists,
2    THEN                                               !
```

```
;    775         0873  3            BEGIN                                       !
;    776         0874  3            name_desc [0] = .nam_block [nam$b_rsl];     ! then fill in the resultant name length
;    777         0875  3            name_desc [1] = .nam_block [nam$l_rsa];     ! and address.
;    778         0876  3            END
;    779         0877  2        ELSE
;    780         0878  2            IF .nam_block [nam$b_esl] NEQ 0             ! If RMS created an expanded string
;    781         0879  2            THEN                                        ! but couldn't open the file,
;    782         0880  3                BEGIN                                   !
;    783         0881  3                name_desc [0] = .nam_block [nam$b_esl]; ! then fill in the expanded name length
;    784         0882  3                name_desc [1] = .nam_block [nam$l_esa]; ! and address.
;    785         0883  3                END
;    786         0884  2            ELSE                                        ! Otherwise, no RMS name information is available.
;    787         0885  3                BEGIN
;    788         0886  3                name_desc [0] = .infile_desc [dsc$w_length];! So use the file name length
;    789         0887  3                name_desc [1] = .infile_desc [dsc$a_pointer];! and length passed by the CLI.
;    790         0888  2                END;
;    791         0889  2
;    792         0890  2    !
;    793         0891  2    ! Signal the error condition.
;    794         0892  2    !
;    795         0893  2
;    796         0894  2        SIGNAL (                                        ! Signal error with the following arguments:
;    797         0895  2                .message_id,                            !     the message identifier,
;    798         0896  2                1,                                      !     the number of message arguments,
;    799         0897  2                name_desc,                              !     the address of input name descriptor,
;    800         0898  2                .fab_block [fab$l_sts],                 !     the primary RMS completion code,
;    801         0899  2                .fab_block [fab$l_stv]);                !     and the scondary RMS completion code.
;    802         0900  2
;    803         0901  2
;    804         0902  1        END;



                                      0000 00000        .ENTRY  DEL$FILE_ERROR, Save nothing                 ; 0816
                   5E            08     C2 00002        SUBL2   #8, SP                                       ;
                   51      08    AC     D0 00005        MOVL    FAB_BLOCK, R1                                ; 0862
                   50      28    A1     D0 00009        MOVL    40(R1), R0                                   ;
                           03    A0     95 0000D        TSTB    3(R0)                                        ; 0871
                                 08     13 00010        BEQL    1$                                           ;
                   6E      03    A0     9A 00012        MOVZBL  3(R0), NAME_DESC                             ; 0874
            04     AE      04    A0     D0 00016        MOVL    4(R0), NAME_DESC+4                           ; 0875
                                 1B     11 0001B        BRB     3$                                           ; 0871
                           0B    A0     95 0001D  1$:   TSTB    11(R0)                                       ; 0878
                                 0B     13 00020        BEQL    2$                                           ;
                   6E      0B    A0     9A 00022        MOVZBL  11(R0), NAME_DESC                            ; 0881
            04     AE      0C    A0     D0 00026        MOVL    12(R0), NAME_DESC+4                          ; 0882
                                 0B     11 0002B        BRB     3$                                           ; 0878
                   6E    0000'   CF     3C 0002D  2$:   MOVZWL  INFILE_DESC, NAME_DESC                       ; 0886
            04     AE    0000'   CF     D0 00032        MOVL    INFILE_DESC+4, NAME_DESC+4                   ; 0887
                   7E      08    A1     7D 00038  3$:   MOVQ    8(R1), -(SP)                                 ; 0898
                           08    AE     9F 0003C        PUSHAB  NAME_DESC                                    ; 0894
                                 01     DD 0003F        PUSHL   #1                                           ;
                           04    AC     DD 00041        PUSHL   MESSAGE_ID                                   ; 0895
     00000000G     00            05     FB 00044        CALLS   #5, LIB$SIGNAL                               ;
                                        04 0004B        RET                                                  ; 0902
```

```
; Routine Size:  76 bytes,    Routine Base:  $CODE$ + 03E1

;  805          0903  1
```

```
1 ROUTINE condit_handler (signal_array, mechan_array) =
1
1 !++
1 ! Functional description
1 !
1 !       This routine is the condition handler for the main routine.  It
1 !       saves the most severe condition as the exit status.
1 !
1 ! Calling sequence
1 !
1 !       condit_handler (signal_array.ra.v, mechan_array.ra.v)
1 !
1 ! Input parameters
1 !
1 !       signal_array    - the address of the signal array for the condition
1 !       mechan_array    - the address of the mechanism array for the condition
1 !
1 ! Implicit inputs
1 !
1 !       The PURGE_CMD flag in DEL$CLI_STATUS tells whether a DELETE or a PURGE command
1 !               caused this error.
1 !
1 ! Output parameters
1 !
1 !       none
1 !
1 ! Implicit outputs
1 !
1 !       DEL$EXIT_STATUS -  Contains the most severe status encountered.
1 !
1 ! Routine value
1 !
1 !       SS$_RESIGNAL
1 !
1 ! Side effects
1 !
1 !       none
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     signal_array        : REF $BBLOCK;
2
2 BIND
2     signame = signal_array [chf$l_sig_name] : $BBLOCK;  ! Get the condition name
2
2 !
2 ! Update the "most severe error" if the current error is more severe.
2 !
2 IF
2     NOT .signame                                       ! If an error signal
4     AND ((.signame[sts$v_severity]                     ! and severity is worse
4         GTRU .del$exit_status[sts$v_severity])
;  864       0961  3          OR .del$exit_status[sts$v_severity])              ! or no errors yet
;  865       0962  2  THEN
;  866       0963  2      del$exit_status = .signame;                           ! then save it for exit
;  867       0964  2
;  868       0965  2  !
;  869       0966  2  ! If facility number is not that of DELETE (147), then this is a system message, so just resignal.
;  870       0967  2  ! If this is a PURGE command, then signal with the PURGE facility number
;  871       0968  2  ! instead of the DELETE facility number (DELETE is the default prefix).
;  872       0969  2  !
;  873       0970  2
;  874       0971  2  IF .signame [sts$v_fac_no] EQL 147                            ! If facility code says DELETE,
;  875       0972  2    AND .del$cli_status [del$v_purge_cmd]                       !  but this is a PURGE command,
;  876       0973  2  THEN
;  877       0974  2      signame [sts$v_fac_no] = 148;                             !  then signal with PURGE instead of DELETE facility
;  878       0975  2
;  879       0976  2  RETURN SS$_RESIGNAL;                                          ! Resignal to get message
;  880       0977  1  END;



                                                    0004 00000 CONDIT_HANDLER:
                                                                       .WORD    Save R2                                    ; 0904
                                  52      0000'  CF   9E 00002         MOVAB    DEL$EXIT_STATUS, R2                         ;
                      50     04   AC             04   C1 00007         ADDL3    #4, SIGNAL_ARRAY, R0                        ; 0950
                                  12             60   E8 0000C         BLBS     (R0), 2$                                    ; 0958
          51          62          03             00   EF 0000F         EXTZV    #0, #3, DEL$EXIT_STATUS, R1                 ; 0960
          51          60          03             00   ED 00014         CMPZV    #0, #3, (R0), R1                            ;
                                                 03   1A 00019         BGTRU    1$                                          ;
                                  03             62   E9 0001B         BLBC     DEL$EXIT_STATUS, 2$                         ; 0961
                                  62             60   D0 0001E 1$:     MOVL     (R0), DEL$EXIT_STATUS                       ; 0963
00000093  8F     02   A0          0C             00   ED 00021 2$:     CMPZV    #0, #12, 2(R0), #147                        ; 0971
                                                 0F   12 0002B         BNEQ     3$                                          ;
                      0A     EC   A2             03   E1 0002D         BBC      #3, DEL$CLI_STATUS, 3$                      ; 0972
      02  A0          0C          00 00000094    8F   F0 00032         INSV     #148, #0, #12, 2(R0)                        ; 0974
                                  50      0918   8F   3C 0003C 3$:     MOVZWL   #2328, R0                                   ; 0976
                                                 04      00041         RET                                                  ; 0977

; Routine Size:  66 bytes,    Routine Base:  $CODE$ + 042D


;  881       0978  1
;  883          0979  1 END
;  884          0980  0 ELUDOM




                                                                    .EXTRN  LIB$SIGNAL, LIB$STOP

;                                       PSECT SUMMARY
;
;       Name                        Bytes                       Attributes
;
;  $GLOBAL$                                 40  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                                  396  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                                 1135  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)




;                               Library Statistics
;
;                                              -------- Symbols --------       Pages       Processing
;       File                                   Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1            9776       124         1         581        00:02.5




;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:DELEMAIN/OBJ=OBJ$:DELEMAIN MSRC$:DELEMAIN/UPDATE=(ENH$:DELEMAIN)

; Size:          1135 code + 436 data bytes
; Run Time:         00:53.6
; Elapsed Time:     01:13.1
; Lines/CPU Min:     1097
; Lexemes/CPU-Min: 11488
; Memory Used:  153 pages
; Compilation Complete
```

0101 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY